



# LSC @ LDAPCON . 2011



Sébastien Bahloul



## About me

- Developer and software architect
- 10 years experience in IAM
- Recently hired as product manager by a French security editor, Dictao, providing :
  - personal and server signature,
  - certificate and signing validation,
  - electronic vault
  - multi-factor authentication

# Agenda

- Solving one issue : directory synchronization
- The LSC project
- Demonstration
- Open question : how to get updates notification ?

# Handling multiple data sources ?



## Why ?

- *Most of us have already done a directory migration*
- *Who has already written a synchronization script ?  
that has been used once ?*
- *Most of LDAP servers are not providing either a way to synchronize either heterogeneous data or homogeneous data with other implementations*

# Introduction

- Automatic synchronization tools
  - If they already exist, they are quite expensive
    - Directory / database-*specific* replication
    - Application-*specific* connectors (AD, SAP, etc)
  - What about the rest?
    - Between different databases, directories, files?
    - Different data models?
    - Using standards: LDAP, SQL, etc...?

## Goals – functionality

- Read/write to any repository
  - **Database** or **LDAP directory** or ?
  - Standard LDAPv3 operations
  - Connectors for databases
- **Transform** data on-the-fly
  - Adapt to a different data model
  - JavaScript based engine to manipulate data
- Adjustable updates: force values, insert defaults, merge new values with existing ones, no change...

## Goals – usability

- **Quickly** implement a new synchronization
- Highly **configurable**
  - What *exactly* do we read?
  - Powerful **transformations** (correctness is important)
  - What *exactly* do we write?
- Run **fast** (performance is important)
- Easy to setup

=> Fill the gap between the Perl script and the Enterprise ETL



---

# About LSC Project

---

- What is LSC?
  - LDAP Synchronization Connector
  - Open Source project
  - BSD licence
  - Written in Java
  - 6 years in the making
  - 4 years ago *LSC-project.org* created
  - ~10 regular contributors
  
- Website: <http://lsc-project.org>



## LSC : read and write « everywhere »

- Original and best supported connector to LDAP directories
- Additional sources: NIS, database, LDIF/CSV files, Web Services
- Additional destinations: Scripting, database
- Extensible API for custom referential support

## Standards based – Wide support

- Any LDAP server should be supported, tested on:
  - OpenLDAP
  - OpenDS/J
  - Sun DSEE
  - Microsoft Active Directory
  - Novell Directory Services
  - IBM Tivoli Directory Server
- Any database with a JDBC connector, tested on:
  - MySQL, PostgreSQL, Oracle, MSSQL, HSQLDB, ...

# Features

- Full « Refresh » or « RefreshAndPersist » with dryrun support
- On the fly event handling
- Plugin API : connectors, libraries, scripting languages
- JMX and command line remote invocation
- Advanced libraries : encryption, Active Directory, localized strings, ...

## Synchronization rules

- Use your preferred language to write LSC rules!
- LSC built-in and historical support for JavaScript
- Extensible to any JSR 223 compliant language :
  - Php
  - Groovy
  - Unix tools (awk, TCL),
  - Python, Ruby, Scheme (Lisp)
  - ...

# LSC synchronization principles

- First step: sync
  - Get a list of all pivots from the source
  - For each pivot
    - Read the source object
    - Search for the destination object with pivot
    - Build up desired destination object by applying transformations to source object
    - If the destination object exists, calculate modifications
    - Apply: create or modify

# LSC synchronization principles

- Second step: clean (optional)
  - Get a list of all pivots from the destination
  - For each pivot
    - Search for the source object with pivot
    - If the source object doesn't exist, delete from destination
    - Apply: delete
- Alternative step: asynchronous mode
  - Get the next source object to synchronize

# LSC : graphical interface



**ALPHA**



# Demonstration

- Simple use case: synchronize identities
- Involved referential:
  - A source OpenLDAP directory
  - Provisioning to:
    - OpenDJ
    - PostgreSQL

# Roadmap

- Current 2.0 version
  - ✓ Event handling
  - ✓ Write to database
  - ✓ Plugin API
- Next minor version 2.1 (Q1 2012)
  - ✗ Move to a real LDAP API (Apache / OpenDJ LDAP API)
  - ✗ Two-phase commit for file, directory (RFC5805) and database (one-to-many)
  - ✗ Administrative GUI including scheduler
- Next major version 3.0 (later)
  - ✗ Data reconciliation (embedded database)
  - ✗ Many-to-many design

# Try it out! Get involved!

- Main website: <http://lsc-project.org/>
  - Tutorials: quickstart demo
  - Reference documentation

The screenshot shows the homepage of the LSC Project website. At the top left is the LSC Project logo. A navigation bar contains links for ABOUT, DOWNLOAD, DOCUMENTATION, COMMUNITY, ROADMAP, and LOGIN. Below the navigation bar is a dark grey banner with the text: "Ldap Synchronization Connector provides tools to synchronize a LDAP directory from a list of data sources including any database with a JDBC connector, another LDAP directory, flat files..." followed by "Download | Read more...". The main content area features the heading "LDAP SYNCHRONIZATION CONNECTOR (LSC)" and a sub-heading "Database/LDAP/flat files to LDAP configurable engine". There are four main sections with icons: "About LDAP Synchronization Connector (LSC)" (flag icon), "Documentation" (lightbulb icon), "Community" (three people icon), and "Roadmap" (gear icon). A right-hand sidebar contains a "Download" button with a green plus icon, a "Latest Release" section listing "LSC version 1.1 coming soon!" and "Nightly builds available to test", an "Events" section listing "10/07/2009 - RMLL (Nantes)" and "25/06/2009 - LinuxTag (Berlin)", a "Community" section with the text "Get help, contribute or find professional services ... Find out more!", a "Search" section with a search box and a "Search" button, and an "Ohloh statistics" section showing a "ohloh group" icon and "\$304.3K Cost".

## How to get notification updates ?

- The current way of handling:
  - OpenDJ / OpenDS / Oracle / Sun / Netscape : persistent search (draft psearch)
  - Apache DS / OpenLDAP: LDAP Content Synchronization (RFC4533)
- What would be the best way?
  - Ldap Client Update Protocol
  - Per product logs (retro/external/access/...)
  - Application-side database

Thanks for your attention!  
Any questions?